COMPRESSIVE GAUSSIAN MIXTURE ESTIMATION

Anthony Bourrier[⋆†] *Rémi Gribonval*[†] *Patrick Pérez*[⋆]

[⋆] Technicolor, 975 Avenue des Champs Blancs, CS 17616, 35576 Cesson Sévigné, France
[†] INRIA Rennes-Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes, France

ABSTRACT

When fitting a probability model to voluminous data, memory and computational time can become prohibitive. In this paper, we propose a framework aimed at fitting a mixture of isotropic Gaussians to data vectors by computing a low-dimensional *sketch* of the data. The sketch represents empirical moments of the underlying probability distribution. Deriving a reconstruction algorithm by analogy with compressive sensing, we experimentally show that it is possible to precisely estimate the mixture parameters provided that the sketch is large enough. Our algorithm provides good reconstruction and scales to higher dimensions than previous probability mixture estimation algorithms, while consuming less memory in the case of numerous data. It also provides a privacy-preserving data analysis tool, since the sketch doesn't disclose information about individual datum it is based on.

Index Terms— Gaussian mixture estimation, compressive sensing, database sketch, compressive learning.

1. INTRODUCTION

When processing a data collection for learning or for other tasks, it is often useful to fit a probability model to the data to get a more compact representation of the data. The fitted model can be used to represent data in a concise way, to feed learning algorithms that work on densities, to extract features or, simply, to uncover underlying structures.

However, fitting a mixture model often requires extensive access to the data to estimate parameters through the iterations of an optimization algorithm. Depending on the size of the data, this may not be possible: to compute the estimation while meeting the memory and computational limits of the hardware, one must typically randomly subsample the data.

In this paper, we consider the idea of computing a *sketch* of the data, which we define as a concatenation of empirical moments computed in one pass over the data. Given this compressed representation, we wish to estimate a model that fits the data. Figure 1 illustrates this idea. Related works that define sketches of data collections are discussed in section 2.

In order to solve our estimation problem, we search for a model which has a sketch close to the sketch of the data collection. Section 3 is dedicated to the definition of our problem with a special focus on Gaussian Mixture Models (GMM). To solve it, we exploit the analogy of the problem to compressive sensing and derive in section 4 an algorithm analog to Iterative Hard Thresholding (IHT) [1].

We show experimentally in section 5 that it is possible to estimate precisely the GMM density in some cases which will be described later, with a lower memory consumption than EM. We give conclusions and discuss outlooks in section 6.

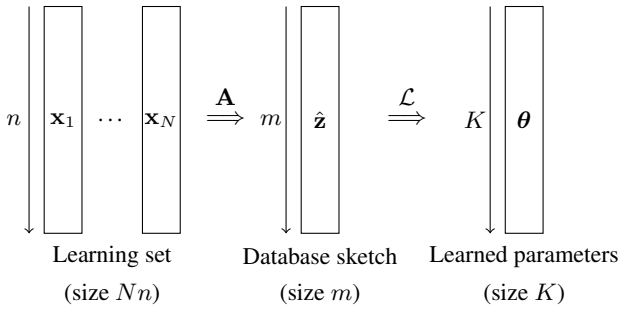

Fig. 1. Illustration of the proposed framework. The learning set is compressed to a sketch $\hat{\mathbf{z}}$ by a sketching operator $\mathbf{A}$ and the mixture parameters $\boldsymbol{\theta}$ are estimated by a learning algorithm $\mathcal{L}$ using only the sketch. Intuitively, we should have $K \leq m$ for the algorithm to work and $m \ll Nn$ for the method to allow computational and memory gains.

2. RELATED WORK

Algorithms using sketches can be found in the database literature [2, 3]. In this case, a sketch is a compressed representation of the data and can be updated whenever an element is added or removed from the database without reprocessing all the data. A popular application is the search for frequent items in a data stream, also called *heavy hitters* [4].

Histogram fitting using a sketch has been explored [5] in the case of n-dimensional discrete vectors. In this case, the sketch is an accumulated random projection of the vectors. This method does not scale to high dimensions, the construction of the histogram from the sketch having complexity which is exponential in n.

Inverse problems on density mixtures have also been studied [6, 7]. Given data drawn from a mixture of candidate functions, both papers propose to cast the reconstruction as the optimization of a sparsity-inducing cost function on the vector of mixture coefficients. Both methods require the considered set of candidate densities to be finite and the elements of this set to be incoherent, *i.e.*, different from each other. These assumptions do not hold for GMM: the centroids of the Gaussians can vary continuously and be arbitrarily close to one another.

Compressed representations of data vectors based on random projections for linear classification have been studied in [8, 9]. These compressed representations aim at replacing a vector $\mathbf{x}$ by $\mathbf{M}\mathbf{x}$ where $\mathbf{M}$ is a dimensionality-reducing matrix, but do not compress the whole data set to a size that is independent of the number N of vectors in the set.

In [10], a compressed representation of sparse probability distributions over multidimensional binary vectors is studied. The com-

pressed representation is obtained by measuring the underlying distribution through random perceptrons and shown to carry enough information for one to be able to reconstruct the initial distribution with a ℓ_1 minimization algorithm.

3. PROBLEM STATEMENT

Let $F \subset L^1(\mathbb{R}^n)$ be a set of probability densities, *i.e.*, $\forall f \in F, f$ is positive and $||f||_1 = \int_{\mathbb{R}^n} f(\mathbf{x})d\mathbf{x} = 1$. Let $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subset \mathbb{R}^n$ be vectors drawn *i.i.d.* from a mixture $p = \sum_{s=1}^k \alpha_s f_s$, with $\forall s, f_s \in F, \alpha_s \geq 0$ and $\sum_{s=1}^k \alpha_s = 1$. We want to find α_s and f_s from $\mathcal{X}$.

Given $\sigma > 0$, we consider the case where F is a family of isotropic Gaussian densities of covariance matrix $\sigma\mathbf{I}$, that is:

$$F = \left\{ f_{\boldsymbol{\mu}} : \mathbf{x} \mapsto \frac{1}{(2\pi)^{\frac{n}{2}}\sigma^n} \exp\left(\frac{\|\mathbf{x}-\boldsymbol{\mu}\|_2^2}{2\sigma^2}\right), \boldsymbol{\mu} \in \mathbb{R}^n \right\}. \quad (1)$$

Given m frequencies $\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_m$, the sketching operator $\mathbf{A}$ is defined as:

$$\langle F \rangle \to \mathbb{C}^m \quad (2)$$

$$q \mapsto (h_1(q), \dots, h_m(q)), \quad (3)$$

where $h_j(q) = \int_{\mathbb{R}^n} q(\mathbf{x}) e^{-i\langle \mathbf{x}, \boldsymbol{\omega}_j \rangle} d\mathbf{x}$ and $\langle F \rangle$ is the linear span of F. The sketch of a density mixture q is thus obtained by sampling the characteristic function of q at m frequencies. Given the empirical data $\mathcal{X}$ drawn from p, an empirical estimate of $\mathbf{A}p$, noted $\hat{\mathbf{z}}$, is given by $\hat{\mathbf{z}} = \left(\hat{h}_1(\mathcal{X}), \dots, \hat{h}_m(\mathcal{X})\right)$ with $\hat{h}_j(\mathcal{X}) = \frac{1}{N}\sum_{i=1}^N e^{-i\langle \mathbf{x}_i, \boldsymbol{\omega}_j \rangle}$.

Reconstructing p as a k-sparse mixture of functions in F can be cast as the following minimization problem:

$$\hat{p} = \underset{q \in \Sigma_k}{\operatorname{argmin}} \|\hat{\mathbf{z}} - \mathbf{A}q\|_2^2, \quad (4)$$

where $\Sigma_k := \{f = \sum_{s=1}^k \alpha_s f_s | \forall s, f_s \in F \wedge \alpha_s \geq 0\}$. Note that we do not enforce the condition $\sum_{s=1}^k \alpha_s = 1$. In practice, we will see that the parameters can still be retrieved only with the positivity constraint.

This minimization problem is reminiscent of compressive sensing, where one wants to reconstruct a sparse finite-dimensional vector from a number of measures that is lower than its dimension. Typically, these measures are linear combinations of the entries of the vector. The difference is that here, the vector is infinite-dimensional and is sparse in a basis (F) which is extremely coherent, *i.e.*, composed of vectors that can be very close to each other (if $\boldsymbol{\mu} \approx \boldsymbol{\nu}$, then $f_{\boldsymbol{\mu}} \approx f_{\boldsymbol{\nu}}$).

4. ALGORITHM

To address the estimation problem (4), we propose an algorithm analogous to Iterative Hard Thresholding (IHT) [1].

4.1. Iterative Hard Thresholding

IHT is designed to reconstruct a k-sparse signal $\mathbf{x}$ of dimension n from m measurements given by $\mathbf{y} = \mathbf{M}\mathbf{x}$, where $\mathbf{M}$ is a $m \times n$ matrix (with $m < n$). At each iteration, IHT updates an estimate $\hat{\mathbf{x}}$ of $\mathbf{x}$, decreasing the objective function $\phi : \hat{\mathbf{x}} \mapsto \frac{1}{2}\|\mathbf{y} - \mathbf{M}\hat{\mathbf{x}}\|_2^2$ while ensuring the k-sparsity of $\hat{\mathbf{x}}$. This is performed in two steps:

1. The n-dimensional gradient of ϕ, noted $\mathbf{g}$, is computed.
2. The update is given by $\hat{\mathbf{x}} \leftarrow H_k(\hat{\mathbf{x}} - \lambda\mathbf{g})$, where λ is a descent step and H_k is a hard thresholding operator which keeps only the k entries of the vector with largest module and sets the others to 0.

4.2. Continuous case

In our case, the signal to reconstruct is p and our estimate $\hat{p}$ is parametrized by a vector $\hat{\boldsymbol{\alpha}} \in \mathbb{R}^k$ of positive weights and by the support $\hat{\Gamma} = \{\hat{\boldsymbol{\mu}}_1, \dots, \hat{\boldsymbol{\mu}}_k\} \subset \mathbb{R}^n$ corresponding to the means of the current estimated Gaussians. The current residual is defined by $\hat{\mathbf{r}} = \hat{\mathbf{z}} - \mathbf{A}\hat{p}$.

Some differences with IHT require a modification of the procedure. In IHT, the signal one wants to reconstruct is supposed sparse in a finite base of vectors, and the gradient of the objective function thus have a finite number of entries which measure the infinitesimal shift of the objective function if a certain entry of the vector is shifted. In our case, the density belongs to the infinite-dimensional space $\langle F \rangle$, and there are as many directions in which the current estimation of the density can be shifted as the cardinality of $\mathbb{R}^n$. The equivalent of the gradient is the function:

$$\begin{aligned} g_{\hat{p}}(\boldsymbol{\mu}) &= \lim_{t \to 0} \frac{1}{2t}\left(\|\hat{\mathbf{z}} - \mathbf{A}(\hat{p} + t f_{\boldsymbol{\mu}})\|_2^2 - \|\hat{\mathbf{z}} - \mathbf{A}\hat{p}\|^2\right) \\ &= -\langle \mathbf{A}f_{\boldsymbol{\mu}}, \hat{\mathbf{r}} \rangle. \end{aligned} \quad (5)$$

Instead of seeking k global extrema as in IHT, we seek a certain number M of local minima of $g_{\hat{p}}$ by initializing randomly a gradient descent algorithm. The random initialization will be discussed in section 5. These local minima are added to the current support $\hat{\Gamma}$ and the hard thresholding is performed by selecting the highest k components of the projection of $\hat{\mathbf{z}}$ on the cone generated by $\hat{\Gamma}$, *i.e.*, the set of linear combinations of elements of $\hat{\Gamma}$ with positive coefficients.

Since the family F is extremely coherent, the local minima we find may be shifted from the true mean vectors because of the imprecision induced by the other components of the mixture. This is why we add a gradient descent step after the search for local minima: this way we reduce further the objective function after having found the local minima.

Algorithm 1 describes the overall procedure. It mainly consists of three steps by iteration:

1. M local minima of $\boldsymbol{\mu} \mapsto g_{\hat{p}}(\boldsymbol{\mu})$ are sought with a gradient descent algorithm with random initialization and are added to the current support.
2. The sketch $\hat{\mathbf{z}}$ is projected on this support with a positivity constraint on the coefficients. Only the k highest coefficients and the corresponding vectors of the support are kept.
3. A gradient descent algorithm is applied to further decrease the objective function with respect to the weights and support vectors.

Algorithms 2, 3, 4 and 5 describe the subfunctions.

4.3. Memory usage

Let's consider that n, k and m are much larger than 1. If we suppose that optimization algorithms only use first-order quantities, their memory costs are dominated by $\mathcal{O}(kn)$. The computation of the cost function of Algorithm 5 requires $\mathcal{O}(km)$. The storage of the operator $\mathbf{A}$ (via the frequencies $\boldsymbol{\omega}_j$) requires $\mathcal{O}(mn)$.

Therefore, the total memory usage is $\mathcal{O}((k+n)m + kn)$ and does not depend on the number N of vectors. In comparison, the

Algorithm 1 Compressive isotropic Gaussian mixture parameter estimation

Input: Sketch $\hat{\mathbf{z}}$, operator $\mathbf{A}$, target sparsity k, integer M.
Initialize $\hat{\Gamma} = \emptyset, \hat{\mathbf{r}} = \hat{\mathbf{z}}$.
repeat
  Set $\boldsymbol{\nu}_1, \dots, \boldsymbol{\nu}_M \leftarrow \text{Find\_min}(\mathbf{A}, \hat{\mathbf{r}}, M)$.
  Set $\hat{\Gamma}' \leftarrow \hat{\Gamma} \cup \{\boldsymbol{\nu}_1, \dots, \boldsymbol{\nu}_M\}$.
  Set $\hat{\boldsymbol{\alpha}}' \leftarrow \text{Proj\_cone}(\hat{\mathbf{z}}, \hat{\Gamma}')$.
  Set $\hat{\boldsymbol{\alpha}}, \hat{\Gamma} \leftarrow \text{Hard\_threshold}(\hat{\boldsymbol{\alpha}}', \hat{\Gamma}', k)$.
  Set $\hat{\boldsymbol{\alpha}}, \hat{\Gamma} \leftarrow \text{Shift\_support}(\hat{\mathbf{z}}, \hat{\boldsymbol{\alpha}}, \hat{\Gamma}, k)$.
  Set $\hat{\mathbf{r}} \leftarrow \hat{\mathbf{z}} - \sum_{j=1}^{k} \hat{\alpha}_j \mathbf{A} f_{\hat{\boldsymbol{\mu}}_j}$.
until Stopping criterion is satisfied
Return $\hat{\boldsymbol{\alpha}}, \hat{\Gamma}$.

Algorithm 2 Find_min($\mathbf{A}, \hat{\mathbf{r}}, M$)

For $i = 1$ to M
  Find a local minimum $\boldsymbol{\nu}_i$ of the function:
  $\boldsymbol{\nu} \in \mathbb{R}^n \quad \mapsto \quad -\langle \mathbf{A} f_{\boldsymbol{\nu}}, \hat{\mathbf{r}} \rangle$
  with a gradient descent algorithm, initialized randomly.
End For
Return $\boldsymbol{\nu}_1, \dots, \boldsymbol{\nu}_M$.

Algorithm 3 Proj_cone($\mathbf{v}, \Gamma = \{\mathbf{u}_1 \dots \mathbf{u}_K\}$)

Solve the following convex optimization problem:
  $\mathbf{a} = \underset{\boldsymbol{\beta} \in \mathbb{R}_+^K}{\text{argmin}} ||\mathbf{v} - \mathbf{U}\boldsymbol{\beta}||_2$, with $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_K]$.
Return $\mathbf{a}$.

Algorithm 4 Hard_threshold($\mathbf{a}, \Gamma = \{\mathbf{u}_1, \dots, \mathbf{u}_K\}, k$)

Let $a_{i_1}, \dots, a_{i_k}$ be the k highest entries of $\mathbf{a}$.
Return $(a_{i_1}, \dots, a_{i_k}), \{\mathbf{u}_{i_1}, \dots, \mathbf{u}_{i_k}\}$.

Algorithm 5 Shift_support($\hat{\mathbf{z}}, \boldsymbol{\alpha}, \Gamma = \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k\}, k$)

Find a local minimum $(\boldsymbol{\alpha}', \boldsymbol{\mu}'_1, \dots, \boldsymbol{\mu}'_k)$ of the function:
  $\mathbb{R}^k \times (\mathbb{R}^n)^k \quad \to \quad \mathbb{R}_+$
  $(\boldsymbol{\beta}, \boldsymbol{\nu}_1, \dots, \boldsymbol{\nu}_k) \quad \mapsto \quad \|\hat{\mathbf{z}} - [\mathbf{A} f_{\boldsymbol{\nu}_1}, \dots, \mathbf{A} f_{\boldsymbol{\nu}_k}]\boldsymbol{\beta}\|_2$,
  using a gradient descent algorithm initialized at
  $\boldsymbol{\alpha}, \boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k$.
Return $\boldsymbol{\alpha}', \{\boldsymbol{\mu}'_1, \dots, \boldsymbol{\mu}'_k\}$.

memory requirement of EM is $\mathcal{O}(N(k+n))$ to store both the vectors and their probabilities to belong to each current component of the mixture. The compressed algorithm allows to memory savings as soon as $m + \frac{kn}{k+n} \lesssim N$. Since $kn \lesssim m$, this condition is nearly equivalent to $m \lesssim N$.

5. EXPERIMENTS

5.1. Experimental setup

To evaluate the behavior of our algorithm, we conducted experiments on vectors drawn from a mixture of k isotropic Gaussians with identity covariance matrices ($\sigma = 1$). In each case, we drew weights uniformly on the simplex[1], and we chose the Gaussian means by drawing random vectors, each entry being drawn from a probability law of density $\mathcal{N}(0, 1)$.

The experiments were performed in the following way: after the choice of the probability distribution p, we drew N random vectors from this probability distribution and computed the empirical sketch of the distribution in one pass of the data, which was then discarded from hard memory (see section 5.2). We then applied the reconstruction algorithm to the sketch to get an approximated mixture $\hat{p}$. The random initialization for the reconstruction algorithm is detailed in section 5.3.

To measure the quality of the reconstruction, we considered a symmetrized version of the Kullback-Leibler (KL) divergence between the true mixture and the estimated one, defined by $D(p||\hat{p}) + D(\hat{p}||p)$, where D denotes the regular KL divergence. To approximate this divergence, we drew $N' = 10^5$ points $(\mathbf{y}_i)_{i=1}^{N'}$ *i.i.d.* from p and computed $\frac{1}{N'} \sum_{i=1}^{N'} \left[\ln\left(\frac{p(\mathbf{y}_i)}{\hat{p}(\mathbf{y}_i)}\right) + \frac{\hat{p}(\mathbf{y}_i)}{p(\mathbf{y}_i)} \ln\left(\frac{\hat{p}(\mathbf{y}_i)}{p(\mathbf{y}_i)}\right)\right]$. We also considered an approximated Hellinger distance, measured by the quantity $1 - \frac{1}{N'} \sum_{i=1}^{N'} \sqrt{\frac{\hat{p}(\mathbf{y}_i)}{p(\mathbf{y}_i)}}$. The KL divergence ranges from 0 to $+\infty$ while the Hellinger distance ranges from 0 to 1. In both cases, lower values mean closer distributions.

[1]We also performed experiments where all the weights were equal to $\frac{1}{k}$ and this didn't alter the conclusions drawn from the experiments.

5.2. Choice of the frequencies

Note that if p is a mixture of probability densities taken in F, then its Fourier transform decays as $\exp\left(-\frac{\sigma^2}{2}\|\boldsymbol{\omega}\|_2^2\right)$, giving an upper bound for the entries of the sketch.

Since we do not want the elements of the sketch to be too small (they should be informative about p), we chose the frequencies $\boldsymbol{\omega}_j$ randomly with a probability law $\mathcal{N}(0, \frac{1}{\sigma^2}\mathbf{Id})$ (thus proportional to the upper bound we found for the values of the sketch). Here, since $\sigma = 1$, this law was $\mathcal{N}(0, \mathbf{Id})$.

5.3. Heuristic for random initialization

The search for local minima in algorithm 2 was initialized randomly by exploiting a measure performed during the construction of the sketch: during the single pass on the data, the norms of the vectors are computed and the maximum of the norms $R = \max_{\mathbf{x} \in \mathcal{X}} ||\mathbf{x}||_2$ is computed. This measure has a negligible effect on the computation time of the sketch and delimits a ball in which the centers of the Gaussians are very probably contained.

We performed the random initialization by drawing a direction uniformly on the unit sphere and multiplying this unit vector by a scalar uniformly drawn in $[0; R]$.

5.4. Results

Figure 2 visually illustrates the behavior of the algorithm on a simple mixture of 4 Gaussians in dimension 2. $N = 10^3$ points were drawn from this mixture and used to compute a $m = 30$-dimensional sketch. As shown in the figure, the parameters are precisely estimated without referring to the initial data. The symmetric KL divergence and Hellinger distance are respectively 0.026 and 0.003.

Figure 3 illustrates the reconstruction quality of our algorithm in dimension 10 for different values of mixture components k and sketch sizes m in terms of Hellinger distance. For each sketch size m ranging from 200 to 2000 with range 200, k was chosen to range from $m/200$ to $m/10$ with step $m/200$. For each choice of parameters, 10 experiments were performed and the depicted value

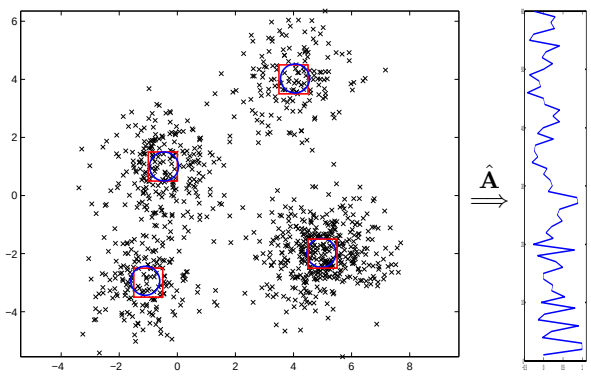

Fig. 2. Real and reconstructed centroids (respectively represented as circles and squares) of 4 Gaussians in dimension 2 from 10^3 points drawn from the mixture. To estimate the 12 real parameters of the mixture, the data was compressed to a complex-valued sketch of dimension 30, represented to the right as a 60-dimensional real signal.

N	Compressed		
	KL div.	Hell.	Mem.
10^3	0.68 ± 0.28	0.06 ± 0.01	0.6
10^4	0.24 ± 0.31	0.02 ± 0.02	0.6
10^5	0.13 ± 0.15	0.01 ± 0.02	0.6
N	EM		
	KL div.	Hell.	Mem.
10^3	0.68 ± 0.44	0.07 ± 0.03	0.24
10^4	0.19 ± 0.21	0.01 ± 0.02	2.4
10^5	0.13 ± 0.21	0.01 ± 0.02	24

Table 1. Comparison between our compressed estimation algorithm and an EM algorithm in terms of precision of the estimation and memory usage (in megabytes). Experiments were performed with $n = 20$, $k = 10$, $m = 1000$. In each cell, the value is a median on 10 experiments with the standard deviation for the precision measures.

is the Hellinger distance such that 80% of the experiments lead to a smaller Hellinger distance. We can essentially observe a gradually increasing measure of the Hellinger distance as the number of mixture components rises. For the considered parameters range, choosing $m = 10kn$, *i.e.*, choosing m so that it contains 10 times more values than the number of parameters to estimate, leads to a Hellinger distance smaller than 0.03 for 80% of the cases.

Table 1 compares our algorithm with a standard EM algorithm in the case where $n = 20$, $k = 10$, $m = 1000$ for values of dataset size N ranging from 10^3 to 10^5. For each case, we can see that the precision of the estimation increases with the number of samples. In the compressed case, this can be explained by the fact that the components of the sketch are better estimated with more points. We notice that the memory used for EM is proportional to the number N of samples in the dataset, while the memory required by the compressed algorithm does not depend on this parameter, which leads to a substantial improvement in memory usage for $N \geq 10^4$. Even with this reduced memory cost, the compressed algorithm is able to provide a precision comparable to the precision of the EM algorithm.

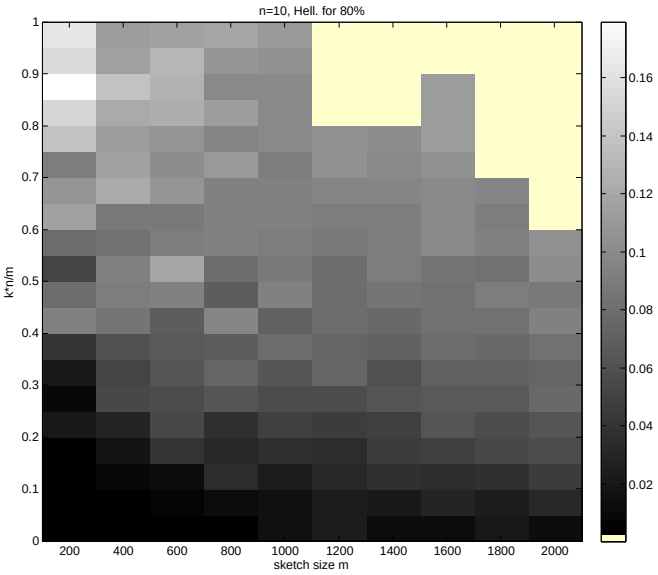

Fig. 3. Quality of reconstruction in dimension $n = 10$, with $N = 10^4$ points, measured as a Hellinger distance (darker is better). Each square corresponds to 10 experiments, and the depicted values are the values of the Hellinger distance under which 80% of performed experiments are placed.

6. CONCLUSION AND OUTLOOKS

We proposed a framework for density estimation exploiting a *sketch* of the data instead of the data itself. This sketch is computed in one pass of the data, which can be discarded on the fly, leading to memory savings in the case of numerous data and to privacy-keeping estimation, since the individual data points cannot be recovered from the sketch. We detailed how this framework can be instantiated for the estimation of a mixture of isotropic Gaussians by deriving an algorithm by analogy with IHT. We experimentally showed that it provides precise reconstruction compared to standard EM while having memory requirements that do not depend on the number of data vectors.

Future work includes generalization to non-isotropic Gaussians, richer choice of sketching functions and other choices of initialization methods for the search of local minima, as well as a theoretical understanding of the well-posedness and stability of the considered problems.

On the experimental side, it would be interesting to see how other methods for large-scale learning such as online EM [11] or stochastic gradient [12] can also be compared with our framework.

Acknowledgments

This work was supported in part by the European Research Council, PLEASE project (ERC-StG-2011-277906).

7. REFERENCES

[1] T. Blumensath and M. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[2] A. C. Gilbert, Y. Kotidis, S. Muthukrishnan, and M. J. Strauss, "How to summarize the universe: Dynamic maintenance of quantiles," in *Conference on Very Large Databases*, 2002.

[3] G. Cormode and S. Muthukrishnan, "An improved data stream summary: The count-min sketch and its applications," *Journal of Algorithms*, vol. 55, no. 1, pp. 58–75, 2005.

[4] G. Cormode and M. Hadjieleftheriou, "Methods for finding frequent items in data streams," *VLDB Journal*, vol. 19, no. 1, pp. 3–20, 2010.

[5] N. Thaper, S. Guha, P. Indyk, and N. Koudas, "Dynamic multidimensional histograms," in *ACM SIGMOD International conference on Management of data*, 2002.

[6] F. Bunea, A. Tsybakov, M. Wegkamp, and A. Barbu, "Spades and mixture models," *Annals of Statistics*, vol. 38, no. 4, pp. 2525–2558, 2010.

[7] K. Bertin, E. Le Pennec, and V. Rivoirard, "Adaptive Dantzig density estimation," *Annales de l'Institut Henri Poincaré (B) Probabilités et Statistiques*, vol. 47, no. 1, pp. 43–74, 2011.

[8] R. Calderbank, R. Schapire, and S. Jafarpour, "Compressed learning : Universal sparse dimensionality reduction and learning in the measurement domain," *Preprint*, 2009.

[9] O.-A. Maillard and R. Munos, "Compressed least-squares regression," in *Neural Information Processing Systems Conference*, 2009.

[10] X. Pitkow, "Compressive neural representation of sparse, high-dimensional probabilities," in *Neural Information Processing Systems Conference*, 2012.

[11] P. Liang and D. Klein, "Online EM for unsupervised models," in *Conference of the North American Chapter of the Association for Computational Linguistics*, 2009.

[12] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning," in *Neural Information Processing Systems Conference*, 2008.